

Web-Js-5

2019년 3월 27일 수요일 오전 10:30

Networking

- AJAX: Request 보내기
- AJAX: Response 받아서 처리하기
- JSON
- AJAX + JSON

• AJAX (Asynchronous JavaScript and XML)

: 브라우저에서 페이지를 이동하지 않고 자바스크립트를 통해 HTTP Request를 보내고 그 응답을 받아 처리할 수 있는 기술
↳ 사용자에게 더 나은 사용 경험 제공, 대부분의 웹사이트에서 사용되고 있는 기술.

1. AJAX를 위한 객체 생성

```
var req = new XMLHttpRequest();  
// HTTP요청을 만들 수 있는 새로운 객체를 생성하는 명령.
```

2. 요청의 방식과 URL 설정

```
req.open("GET", ". / data.txt");  
// http request method와 URL 설정
```

3. 요청 전송

```
req.send();
```

* 응답 확인

- req.response에 저장됨
- 비동기 방식으로 진행하기 때문에 send 메소드 호출 후 바로 코드에서 접근하면 데이터가 비어있음.
- AJAX 진행에 따라 호출되는 callback 함수를 활용해야 함

- 브라우저 옵션

`--disable-web-security` 옵션

: 브라우저의 보안 정책을 우회하기 위해 사용하는 옵션

(설정해줘야!)

--disable-web-security | 끄기

: 브라우저의 보안정책을 우회하기 위해 사용하는 옵션
Same-origin-policy 등을 비활성화함

• Response 받아서 처리하기

- readyState 속성 (0 ~ 4)

0 : open 메소드 호출전

1 : open 메소드 호출후, send 메소드 호출전

2 : 보낸 요청에 대해 응답헤더가 수신된후

3 : 응답의 바다가 수신중일때

4 : 모든응답이 수신되었을때.

- onreadystatechange 속성

: readyState가 변할때마다 호출되는 콜백함수

- Status 속성

: HTTP response의 응답 헤더에 기록된 코드

200 : OK

404 : NOT Found

500 : Internal Err

:

- 응답을 정상적으로 수신한 경우

· ready state : 4

· Status : 200

- 기타 callback function 활용 가능한 속성.

· onloadstart

· onprogress

· onabort

· onerror

· onload

· ontimeout

· onloadend

• JSON (JavaScript object Notation)

• JSON (JavaScript object Notation)

: 자바스크립트의 객체를 문자열로 표현하는 방법

Server → browser [표현] (문서이전용)



- JSON.stringify(Object)

: 인자로 받은 객체를 JSON 문자열로 변환

- JSON.parse(String)

: 인자로 받은 문자열을 Javascript object로 변경하여 반환함

(undefined와 메소드는 변환x ⇒ null값)

example

```
Var arr = [ "배열안들어", 1234, true, null, undefined ]
```

```
Var t = JSON.stringify(arr);
```

(t 가하면 "배열안들어")

JSON.parse(t): (가하면 배열)

```
> var arr = ["배열", 1234, undefined, true];
```

```
< undefined
```

```
> var t = JSON.stringify(arr);
```

```
< undefined
```

```
> JSON.parse(t);
```

```
✖ Uncaught SyntaxError: Unexpected identifier
```

```
> JSON.parse(t);
```

```
< ▶ (4) ["배열", 1234, null, true]
```

```
> var obj = {  
  "str" : "배열",  
  "num" : 1234,  
  "undefined": undefined,  
  "boolean" : true  
}
```

```
< undefined
```

```
> var t2 = JSON.stringify(obj);
```

```
< undefined
```

```
> t2
```

```
< '{"str":"배열","num":1234,"boolean":true}'
```

```
> JSON.parse(t2);
```

```
< ▶ {str: "배열", num: 1234, boolean: true}
```

```
> JSON.parse(t2);  
< ▶ {str: "배열", num: 1234, boolean: true}
```

• AJAX + JSON

```
1 <html>  
2 <head>  
3 <meta charset = "utf-8">  
4 <style>  
5 .character {color : black;}  
6 .word{color : gray;}  
7 </style>  
8 <script>  
9 var req = new XMLHttpRequest();  
10 req.onreadystatechange = function a(){  
11 console.log(this.readyState, this.status);  
12 if(this.readyState == 4){  
13 console.log(this.response);  
14 var data = JSON.parse(this.response); // 배열형태의 데이터임  
15 for(var i in data){  
16 var t = document.getElementById("template").cloneNode(  
17 true);  
18 t.removeAttribute("id"); 삭제  
19 t.children[0].innerText = data[i].id; id값 msg 가져와  
20 t.children[1].innerText = data[i].msg;  
21 document.body.appendChild(t);  
22 }  
23 }  
24 }  
25 req.open("GET", "../lec04_json_data.txt");  
26 req.send();  
27 </script>  
28 </head>  
29 <body>  
30 <p id = "template">  
31 <span class = "character"> 주인공 </span> : <span class = "word  
32 "> 명대사 </span>  
33 </p>  
34 </body>  
35 </html>
```